Col*Fusion: Not Just Jet Another Data Repository

Evgeny Karataev[1] and Vladimir Zadorozhny[1]
[1] School of Information Sciences, University of Pittsburgh

Abstract
In this poster we introduce Col*Fusion – a novel architecture for large-scale data integration, fusion and preservation based on crowdsourcing. Col*Fusion is implemented as easy-to-use web application and provides uniform data submit and integration interface. It provides all functionality expected from professional data archival repository, but also solves two main problems of current approaches – repository and dataset isolation – by involving users into active participation of both data submission and integration processes.
Keywords: data repository, data integration, data preservation, crowdsourcing, open access
Citation: Karataev, E., & Zadorozhny, V. (2014). Col*Fusion: Not Just Yet Another Data Repository. In *iConference 2014 Proceedings* (p. 928–932). doi:10.9776/14320

Acknowledgements: This research was supported by NSF BCS-1244672 grant.
Contact: epk8@pitt.edu, vladimir@sis.pitt.edu

1 Introduction

Research activities in all fields produce data in different forms and shapes. Advances in computing technology allow produced data to be stored in a digital format. In addition, more and more historical records, which originally were captured on paper, are being digitized. The move towards digital data is ubiquitous; it introduces new ways of making the data available to the public and be reused (Borgman, 2008, 2009). Nevertheless, very often research data are not shared at all, or shared on researcher's or university's web page, making it less discoverable (LeClere, 2010; Nelson, 2009). Crosas (2011) argues that researchers are reluctant to share their data because traditional approaches do not facilitate control and ownership of the data by the author. In this poster we identify two other problems with current approaches (Section 2) and introduce our Col*Fusion system as a solution (Section 3).

2 Motivation

A number of tools (e.g., DataUp (Strasser, 2013)) and data repositories (e.g., ONEShare (Strasser, 2013), Dataverse Network (King, 2007), DataDryad (datadryad.org), DSpace (Smith et al., 2003), Socrata (socrata.com), Factual (factual.com)) were developed to facilitate data sharing and preservation processes. Usually a data repository is a cloud service with web interface that allow users to submit their data via browser. Advantages of data repositories include ease of use, persistent storage, public distribution, and recognition (through citation via unique dataset identifier), and search for datasets based on metadata. Some repositories provide visualization tools and statistical analysis. The disadvantages of current approaches include repository isolation and dataset isolation within a repository. The former problem is related to the fact that some repositories are created only for specific research areas, journals or universities. Therefore users would need to know where to find the dataset they are interested in and where to submit their dataset. Dataverse Network and Datalib (datalib.org) attempt to solve the problem by allowing users to search within a set of repositories: the first one does it automatically as all dataverse networks are connected, and the second one allows users to create and curate records that describe data repositories that users can search. The later problem – dataset isolation – to the best of our knowledge is present in all data repositories.

Perhaps, the dataset isolation problem is better shown by an example. Interdisciplinary research, which is becoming more common and more often funded, touches several areas. As the result, for the interdisciplinary research question, a researcher might need to have data that are produced by different researches and stored in separate datasets. Answering her question, researcher would need to manually find all related datasets and then merge them by herself. Suppose that the required data are stored in datasets D_1 and D_2. D_1 and D_2 might not be directly related to each other (they might not share any common variables), however they might be related via other datasets, for example D_1 is related to D_3, D_3 to D_4 and D_4 is related to D_2. Once found, those datasets need to be integrated. Data integration is rather a complex procedure consisting of several activities such as schema matching, record linkage, query execution and search over integrated sources, and keeping track of lineage and provenance. The integration is even more complex if the datasets come in different formats.

The data integration problem has been an interest of both academic and industrial research for the last 30 years. Architectures of current data integration systems vary from warehousing to virtual integration that leave the data at the sources and access it at query time (Doan, Halevy, & Ives, 2012). To address the limitations of top-down approach with one global mediated schema, Peer-to-peer (P2P) (Ng, Ooi, Tan, & Zhou, 2003; Halevy et al., 2004; Wang, Rabsch, Kling, Liu, & Pearson, 2007) and Collaborative Data Sharing Systems (CDSS) (Green et al., 2007; Talukdar, Ives, & Pereira, 2010) have been proposed and developed. However, the disadvantages of traditional data integration systems, P2P, and CDSS systems include long setup and hard to use requiring users to have certain expertise.

3 Col*Fusion

Recognizing the problem, we introduce Col*Fusion – a novel architecture for large-scale data integration, fusion and preservation based on crowdsourcing. Col*Fusion could be though of as an interdisciplinary data repository, however the datasets are not isolated, but connected. In fact, each dataset could be seen as a piece of bigger puzzle that describe world from one perspective. With time Col*Fusion connects the pieces together to complete the puzzle. We have implemented Col*Fusion as web-based application that provides easy-to-use uniform interface for data submission and integration.

Data submission module allows users to submit data from heterogeneous sources and formats, such as Excel, SPSS and CSV files, dump files from MySQL, PostgreSQL and Microsoft SQL databases. In fact, the number of file formats as well as file organization can be expanded by Col*Fusion users. We use Pentaho Data Integration (Casters, Bouman, & Van Dongen, 2010) (aka Kettle) on the back end for extracting, transforming and loading (ETL) data into Col*Fusion repository. Kettle is free and open-sourced, it allows users to specify ETL tasks via intuitive, graphical, drag and drop design environment and save it as a transformation file. Kettle support large number of data sources including leading Hadoop distributions, NoSQL databases, and other big data stores. Col*Fusion users can create a custom Kettle transformation and submit into the Col*Fusion. Some formats are more common than others. Col*Fusion users can share Kettle transformations with other users to handle particular file organization. Therefore most users do not need to do a lot of preparatory work to submit their datasets into Col*Fusion thus makes it easier to use.

Once dataset is submitted, Col*Fusion mine relationships automatically. You can think of Col*Fusion relationships as foreign keys in relational data model. Currently automatic relationship mining algorithm establishes a relationship between two datasets D_1 and D_2 if they share common variables. Relationships can also be added manually by users if Col*Fusion cannot find them automatically due to distinct variable names or if relationship involve mapping of several variables to one (e.g. D_1 might have a date split into three variables, whereas D_2 might have it as one variable).

Each relationship has name, description, and average confidence associated with it (Fig. 1). Confidence values for relationships are provided by Col*Fusion users and basically reflect their believe that relationships hold. Relationships consist of links – the actual connections between variables in datasets.

Each link has two numerical values associated with it that are automatically calculated. The values reflect data overlapping ratios on the both ends of a link. For example, both D_1 and D_2 might have "State" variables that denote political entity forming part of USA. However D_1 might have full names of state whereas D_2 might have state name abbreviations. Without knowledge of mapping between full state names and abbreviates, data overlapping ratios would be 0 and merging D_1 and D_2 would yield an empty dataset.

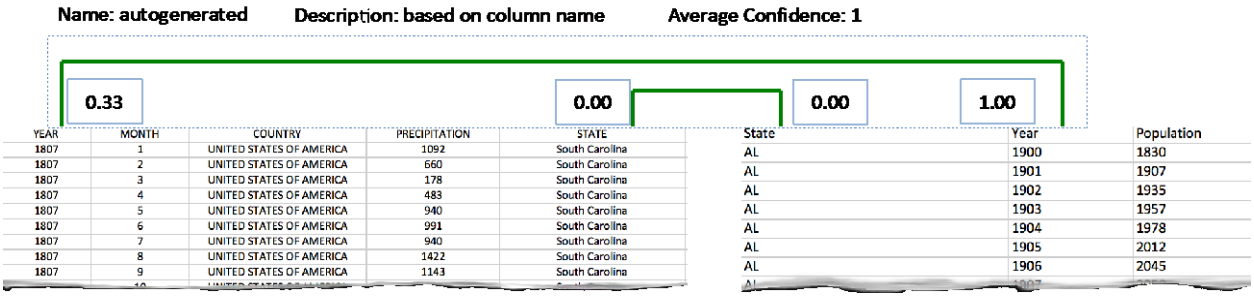

Figure 1: Automatically discovered Col*Fusion relationship between two datasets

Relationship's links can involve data transformation assigned by users (or automatically). One type of transformation is synonyms. Synonyms are used to specify mapping between variables on a value basis (Fig. 2). One of the advantages of this type of transformation is that it is possible to specify many to many mapping. Transformation can also be specified as a transformation function that is applied to each value of a variable. For example, date conversion from DD-MM-YYYY to MM-DD-YYYY format.

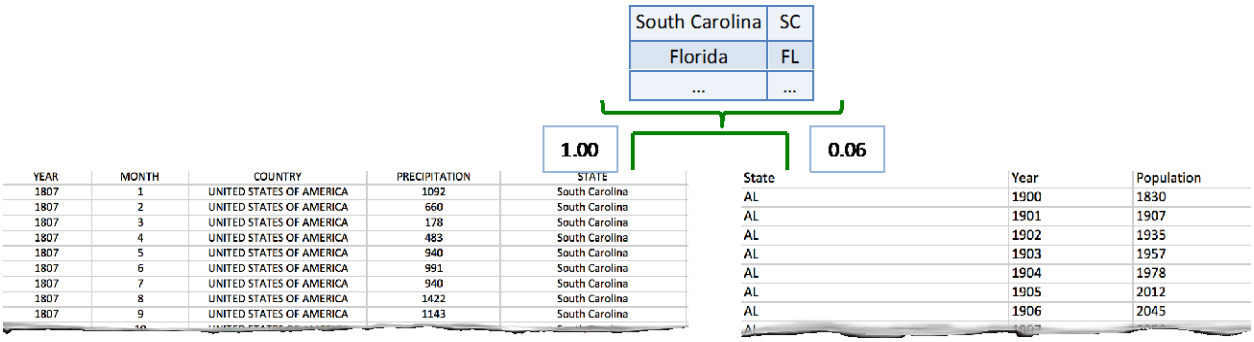

Figure 2: Example of synonyms transformation

One disadvantage of synonyms transformation appears when there is a large number of distinct values that are not matched; especially if those mapping are well know and might be available (e.g., US state mapping is well know). Col*Fusion can deal with such situations by traversing relationships graph (Fig. 3). For example, let D_3 be a datasets that have US state mappings (D_3 might be a results of research which is completely independent from D_1 and D_2 and submitted by other users). Then D_1 and D_2 can be merged via D_3. In general two datasets might be related to each other via several other datasets.

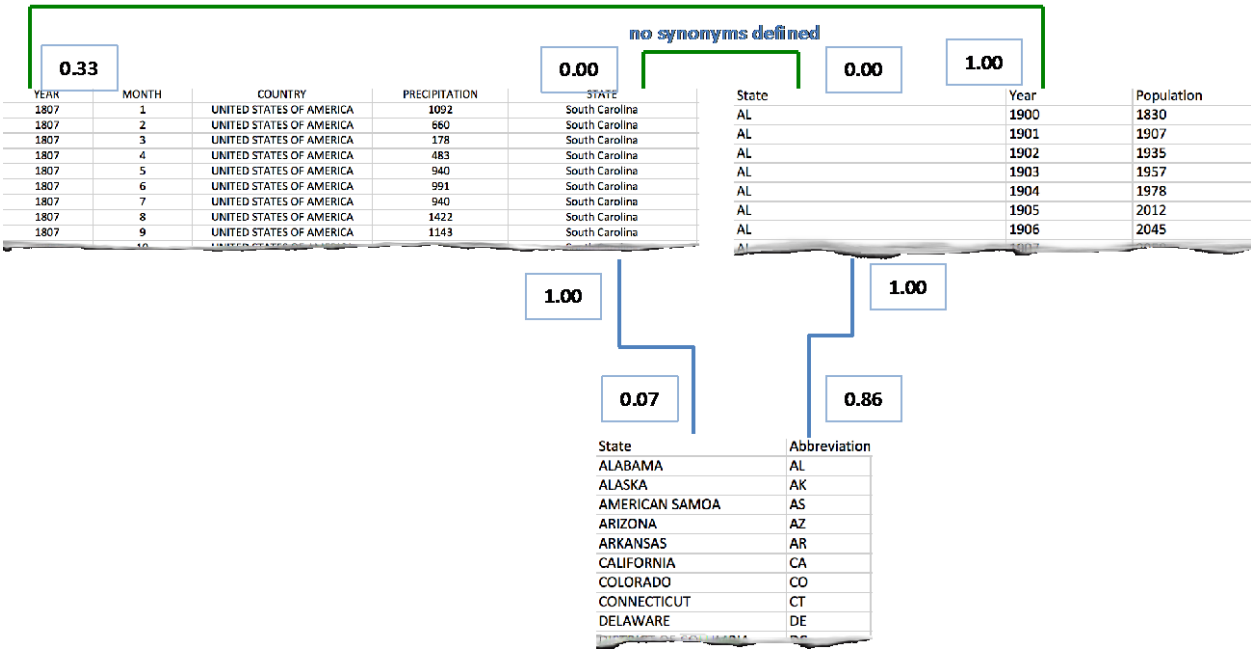

Figure 3: Example of relationship graph traversal to merge datasets

Relationships can be used by users to see how their datasets are related to other datasets and "move" from one dataset to another, but also relationships are used when users perform search in Col*Fusion. Col*Fusion maintains schema graph in which vertices represent data tables and edges represent relationships. When user posts a keyword query, the system performs three steps to answer it. First, Col*Fusion finds all datasets (vertices in the graph) that contain the keywords. Second, it traverses the schema graph to find all paths between the set of vertices found in step one. Third, it translates each path to an SQL query by mapping every vertex to a dataset and every edge to SQL join operator. Therefore, the result of the search is not just a list of datasets that have variables user is interested in, but rather a merged dataset or a list of merged datasets if there are several possible paths to perform the merge. The list is ranked based on relationships' confidence and data overlapping values (e.g., the rank is higher for those paths which have higher average confidence and data overlapping values).

Col*Fusion provides provenance information in OPM format (Moreau et al., 2011) for merged datasets, so users know where each variable came from. In addition, merged datasets can be visualized, downloaded in several formats (regardless of original format) and shared with other users.

4 Conclusion

While the Col*Fusion involves some labor from users, it addresses the dataset isolation problem that has long been resistant to resolution. While datasets can be connected through analysis of the file-level metadata on their sources and overall characteristics, there has not previously been an application that connects the variable-level metadata within datasets. Employing variable-level relationships between datasets allows a third party (users who are not the experts in that area) to reuse the data to cross boundaries, build scientific knowledge and thus advance science.

5 References

Borgman, C. L. (2008). Data, disciplines, and scholarly publishing. *Learned Publishing, 21*(1), 29-38.

Borgman, C. L. (2009). The digital future is now: A call to action for the humanities. *Digital humanities quarterly, 3*(4).

Casters, M., Bouman, R., & Van Dongen, J. (2010). Pentaho Kettle solutions: building open source ETL solutions with Pentaho Data Integration, *Wiley. com*.

Crosas, M. (2011). The dataverse network®: an open-source application for sharing, discovering and preserving data. *D-Lib Magazine, 17*(1).

Doan, A., Halevy, A., & Ives, Z. (2012). Principles of data integration. *Access Online via Elsevier*.

Halevy, A. Y., Ives, Z. G., Madhavan, J., Mork, P., Suciu, D., & Tatarinov, I. (2004). The piazza peer data management system. *Knowledge and Data Engineering, IEEE Transactions on, 16*(7), 787-798.

Green, T. J., Karvounarakis, G., Taylor, N. E., Biton, O., Ives, Z. G., & Tannen, V. (2007). ORCHESTRA: Facilitating collaborative data sharing. *In Proceedings of the 2007 ACM SIGMOD international conference on Management of data,* 1131-1133.

King, G. (2007). An introduction to the Dataverse Network as an infrastructure for data sharing. *Sociological Methods & Research, 36*(2), 173-199.

LeClere, F. (2010). Too many researchers are reluctant to share their data. *The Chronicle of Higher Education*.

Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., ... Myers, J. (2011). The open provenance model core specification (v1. 1). *Future Generation Computer Systems, 27*(6), 743-756.

Nelson, B. (2009). Data sharing: Empty archives. *Nature, 461*(7261), 160-163.

Ng, W. S., Ooi, B. C., Tan, K. L., & Zhou, A. (2003). PeerDB: A P2P-based system for distributed data sharing. In *Data Engineering, 2003. Proceedings. 19th International Conference on,* 633-644.

Smith, M., Barton, M., Bass, M., Branschofsky, M., McClellan, G., Stuve, D., ... Walker, J. H. (2003). DSpace: An open source dynamic digital repository, *D-Lib Magazine 9*(1).

Strasser, C. (2013). DataUp: Enabling data stewardship for researchers. *IConference 2013 Proceedings*, 657-658. doi:10.9776/13300

Talukdar, P. P., Ives, Z. G., & Pereira, F. (2010). Automatically incorporating new sources in keyword search-based data integration. *In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, 387-398.

Wang, F., Rabsch, C., Kling, P., Liu, P., & Pearson, J. (2007). Web-based collaborative information integration for scientific research. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, 1232-1241.

6 Table of Figures